LISTING OF CLAIMS

The list of claims provided below replaces all prior versions and lists of claims in the application. Thus, claims 1-26 are pending.

Please amend the claims as follows.

1. (original):   A method for loop optimization within a dynamic compiler system, comprising:

creating a pre-loop structure based on an original loop structure, wherein the pre-loop structure is capable of testing indexing expressions for underflow;

generating a main loop structure having indexing expressions based on the original loop structure, wherein the indexing expressions cannot produce an underflow, and wherein the indexing expressions cannot produce an overflow; and

creating a post-loop structure based on the original loop structure, wherein the post-loop structure is capable of testing indexing expressions for overflow.

2. (original):   A method as recited in claim 1, wherein the pre-loop structure includes an array boundary test.

3. (original):   A method as recited in claim 2, wherein the post-loop structure includes an array boundary test.

4. (original): A method as recited in claim 3, wherein the main loop structure does not include an array boundary test.

5. (original): A method as recited in claim 1, further including the operation of compiling a computer program during execution of the computer program.

6. (original): A method as recited in claim 5, further including the operation of interpreting lines of the computer program during execution of the computer program.

7. (original): A method for loop optimization within a dynamic compiler system, comprising:

executing a computer program having an original loop structure;

compiling the original loop structure during the execution of the computer program; and

creating a range check elimination loop structure based on the original loop structure during the compiling operation, wherein the range check elimination loop structure includes a pre-loop structure, a main loop structure, and a post-loop structure.

3

8. (original):   A method as recited in claim 7, wherein the pre-loop

structure is capable of testing indexing expressions for underflow.

9. (original):   A method as recited in claim 8, wherein the main loop

structure, wherein indexing expressions included in the main loop structure cannot

produce an underflow, and wherein the indexing expressions cannot produce an

overflow.

10. (original):   A method as recited in claim 9, wherein the post-loop

structure is capable of testing indexing expressions for overflow.

11. (original):   A method as recited in claim 10, wherein the pre-loop

structure includes an array boundary test.

12. (original):   A method as recited in claim 11, wherein the post-loop

structure includes an array boundary test.

13. (original):   A method as recited in claim 12, wherein the main loop

structure does not include an array boundary test.

14. (original):   A method as recited in claim 13, further including the operation of interpreting lines of the computer program during execution of the computer program.

15. (original):   A range check elimination loop structure, comprising:

a pre-loop structure based on an original loop structure, the pre-loop structure capable of testing indexing expressions for underflow;

a main loop structure having indexing expressions based on the original loop structure, wherein the indexing expressions cannot produce an underflow, and wherein the indexing expressions cannot produce an overflow; and

a post-loop structure based on the original loop structure, wherein the post-loop structure is capable of testing indexing expressions for overflow.

16. (original):   A range check elimination loop structure as recited in claim 15, wherein the pre-loop structure includes an array boundary test.

17. (original):   A range check elimination loop structure as recited in claim 16, wherein the post-loop structure includes an array boundary test.

5

18. (original):   A range check elimination loop structure as recited in claim 17, wherein the main loop structure does not include an array boundary test.

19. (original):   A range check elimination loop structure as recited in claim 15, wherein the pre-loop structure, the main loop structure, and the post-loop structure, are generated using a dynamic compiler.

20. (original):   A range check elimination loop structure as recited in claim 19, wherein the dynamic compiler generates the main loop structure, and the post-loop structure during execution of a computer program containing source code for the range check elimination loop structure.

21. (new):   A method for loop optimization, comprising:

creating a pre-loop tree structure based on an original loop structure, wherein the pre-loop tree structure is capable of testing indexing expressions for underflow;

generating a main loop tree structure having indexing expressions based on the original loop structure, wherein the indexing expressions cannot produce an underflow, and wherein the indexing expressions cannot produce an overflow; and

creating a post-loop tree structure based on the original loop structure, wherein the post-loop tree structure is capable of testing indexing expressions for overflow.

22. (new): A method as recited in claim 21, wherein the pre-loop tree structure includes an array boundary test.

23. (new): A method as recited in claim 22, wherein the post-loop tree structure includes an array boundary test.

24. (new): A method as recited in claim 23, wherein the main loop tree structure does not include an array boundary test.

25. (new): A method as recited in claim 21, further including the operation of compiling a computer program during execution of the computer program.

26. (new): A method as recited in claim 25, further including the operation of interpreting lines of the computer program during execution of the computer program.